AF / 2185

Attorney's Docket No.: 10559-292001 / P9299; APD1809-1-US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Ravi P. Singh et al.	Art Unit:	2185
Serial No.:	09/675,569	Examiner:	Midys Rojas
Filed:	September 29, 2000		
Title:	A FIFO WRITE/LIFO READ TRACE BUFFER WITH SOFTWARE AND HARDWARE LOOP COMPRESSION		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

Sir:

Appellant files this Brief on Appeal to perfect the Notice of Appeal filed March 23, 2006.

(1) Real Party in Interest

This case is assigned to Intel Corporation and Analog Devices, Inc. who are hence the real parties in interest.

(2) Related Appeals and Interferences

There are no known related appeals and/or interferences.

(3) Status of Claims

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

June 12, 2006
Date of Deposit

Julie H. Giordano
Signature

Julie H. Giordano
Typed or Printed Name of Person Signing Certificate

Claims 1-7, 9-23, and 27-34 are pending. Claims 8 and 24-26 have been canceled. Claims 13-17 stand allowed. Claims 1-7, 9-12, 18-23 and 27-34 stand rejected under 35 U.S.C. § 103(a) as obvious.

(4) Status of Amendments

All claim amendments have been entered.

(5) Summary of Claimed Subject Matter

This application relates to systems and techniques for trace buffering.

Trace buffering can involve logging the "flow" of a set of data processing instructions. In particular, the path that a program has taken during execution is stored and used to recreate what happened during program execution. Tracing can thus be used to debug the program. *See Specification*, page 8, line 15-18.

During execution, program code is generally executed sequentially. *See Specification*, page 8, line 17-20. However, certain unexpected events can lead to a break in the expected program flow. *See Specification*, page 8, line 18-20. Since execution between such jump events is sequential, the program flow may be reconstructed from a record of the jump events. In order to reduce the memory required for the trace buffer and its speed of operation, the trace buffer may only store the addresses of instructions that caused the program to jump or break, e.g., branch instructions, exceptions, interrupts, and the addresses of the target of the branch. *See Specification*, page 9, line 20 - page 10, line 3.

A branch is one example of a jump event in a program flow. Each branch may be identified by a pair of addresses, a branch target address and a branch source address. *See Specification*, page 10, line 4 - 10.

The inventors have recognized that a program flow may include program loops that lead to such jump events. *See Specification*, page 12, line 18 - 21. Such loops can rapidly fill the trace buffer, increasing the frequency of readout operations and slowing the performance of trace gathering. In particular, while in an n-cycle loop, the stored branch target/source address pairs are repeated n-times and without providing significant information about program flow. *See Specification*, page 12, line 18 - 24.

To address these problems, the inventors have developed systems and techniques for compressing out repeated target/source address pairs. *See Specification*, page 12, line 24 - page 13, line 2.

Claim 1

One such system and technique is the subject of claim 1. Claim 1 relates to a trace buffer circuit. *See, e.g.*, FIG. 5; *Specification*, page 3, line 16-17. The trace buffer circuit includes a plurality of interconnected registers, including a first end register to input and output addresses of fetched instructions during a trace operation, a second end register, and a plurality of middle registers connected between said first end register and said second end register. *See, e.g., FIG. 5*, trace buffer registers tbuf.0 - tbuf.31.

The trace buffer circuit also includes a write path to shift an instruction address in one of said plurality of interconnected registers by two registers toward the second end register on a write operation. *See, e.g., FIG. 5*, write paths 514, 516; *Specification*, page 11, line 6-9.

The trace buffer circuit also includes a first holding register and a second holding register. *See, e.g., FIG. 5*, registers 510, 512.

The trace buffer circuit also includes a first comparator to compare a new branch target address corresponding to a loop in the first holding register to a stored branch target address in the first end register. *See, e.g.*, FIG. 5, comparator 550; *Specification*, page 13, line 12-18.

The trace buffer circuit also includes a second comparator to compare a new branch source address corresponding to the loop in the second holding register to a stored branch source address in a first adjacent register, said first adjacent register being connected to the first end register on a read path. *See, e.g.*, FIG. 5, comparator 550; *Specification*, page 13, line 12-18.

The trace buffer circuit also includes a compression indication circuit to generate a compression indicator in response to the new branch target address matching the stored branch target address and the new branch source address matching the stored branch source address. *See, e.g.*, *Specification*, page 14, line 5-19.

Claim 18

Claim 18 relates to a method that includes performing a trace operation and performing a compression operation. The trace operation includes storing an address pair corresponding to a loop in fetched instructions in a trace buffer. *See, e.g.*, *Specification*, page 10, line 11-15. The compression operation includes comparing the stored address pair to a new address pair in fetched instructions, and setting a least significant bit of an address in the stored address pair in response to the new address pair matching the stored address pair. *See, e.g.*, *Specification*, page 13, line 15-18 and page 14, line 5-8.

Claim 27

Claim 27 relates to an apparatus. The apparatus includes instructions residing on a machine-readable medium for use in a trace buffer. The instructions are operable to a machine to perform a trace operation and perform a compression operation. The trace operation includes storing an address pair corresponding to a loop in fetched instructions in the trace buffer. *See, e.g., Specification*, page 10, line 11-15. The compression operation includes comparing the stored address pair to a new address pair in fetched instructions and setting a least significant bit of an address in the stored address pair in response to the new address pair matching the stored address pair. *See, e.g., Specification*, page 13, line 15-18 and page 14, line 5-8.

(6) Grounds of Rejection to be Reviewed

Appellant requests that the following grounds of rejection be reviewed:

I. whether claim 1 is unpatentable under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 6,094,729 to Mann (hereinafter "Mann"), U.S. Patent No. 5,553,010 to Tanihara et al. (hereinafter "Tanihara"), and U.S. Patent Publication No. 2003/0115424 to Bachand et al. (hereinafter "Bachand"); and

II. whether claims 18 and 27 are unpatentable under 35 U.S.C. § 103(a) as obvious over Mann and Bachand.

(7) Argument

I: REJECTIONS OF CLAIM 1 AND ITS DEPENDENCIES UNDER 35 U.S.C. § 103(a)

Claim 1 was rejected under 35 U.S.C. § 103(a) as obvious over Mann, Tanihara, and Bachand. The rejection contends that it would have been obvious for one of ordinary skill to combine Mann, Tanihara, and Bachand to arrive at the subject matter recited in claim 1. Appellant respectfully disagrees.

In this regard, Mann describes a debug interface that includes a trace buffer 200 that stores entries indicative of the order in which instructions are performed by a processor. *See Mann*, col. 8, line 10-13. Since trace buffer 200 has a limited storage capacity, Mann describes that trace compression is used to limit the amount of data stored at trace buffer 200. *See Mann*, col. 18, line 10-12. As such, Mann is the *only* cited reference that deals with trace compression at all.

Mann's trace compression relies upon screening of the information that is reported in trace data to limit the amount of stored data. In particular, Mann prefers that only a subset of the instruction addresses in the program flow are recorded in trace data. *See, e.g., Mann*, col. 18, line 15-25. For example, Mann explicitly excludes addresses that are otherwise obtainable from a program image. *See Mann*, col. 18, line 15-19.

However, the mere fact that disruption in the program flow has occurred does not mean that Mann records the addresses associated with the disruption. For example, Mann identifies that CALL instructions are disrupting events but they are not reported unless the target address is data dependent. *See Mann*, col. 18, line 26-31. Unconditional branch instructions are compressed in the same manner. *See Mann*, col. 18, line 33-35. Conditional instructions are only reported when the address is not in immediate format. *See Mann*, col. 18, line 36-40.

Moreover, when disruptions in the program flow are reported, only a single target address or segment base address is recorded per disruption. *See, e.g., Mann*, col. 18, line 41-47 (describing that only the target addresses in exception events and return instructions are recorded in the trace record). The table bridging cols. 17-18 makes this clear as it sets forth the exact trace data (TDATA) recorded for each trace code (TCODE). Such limited recording is not surprising, given that Mann identifies that even these single (32-bit) addresses are be recorded in two separate (16-bit) trace data entries in two separate trace data records. *See Mann*, col. 20, line 15-29 (describing that various target addresses are always recorded in 16-bit high order and low order pairs).

As a result of this screening, few of Mann's trace entries contain address values at all. *See, e.g., Mann*, col. 19, line 21. Indeed, on rare occasions, Mann's trace buffer 200 will not contain a single entry with an address. *See Mann*, col. 23, line 37-39. Further, as best understood by Appellant, Mann's trace buffer 200 never includes a source address, much less a source address and a target address for the same single disruption in a program flow.

Against this backdrop, the rejection of claim 1 contends that, on the basis of Bachand's disclosure, it would be obvious for one of ordinary skill to add first and second holding registers; a first comparator to compare a new branch target address corresponding to a loop in the first holding register to a stored branch target address in the first end register; a second comparator to compare a new branch source address corresponding to the loop in the second holding register to a stored branch source address in a first adjacent register, said first adjacent register being connected to the first end register on a read path; and a compression indication circuit to generate a compression indicator in response to the new branch target address matching the stored branch target address and the new branch source address matching the stored branch source address, as recited in claim 1, to Mann's trace buffer 200.

Appellant respectfully disagrees. As discussed above, Mann's trace compression relies upon screening of the information that is recorded in trace buffer 200. As a result of this screening, few of Mann's trace entries even contain address values, much less the stored source and target addresses recited in claim 1. It defies common sense to conclude that one of ordinary skill would somehow be motivated to compare information that is not, in large part, stored.

The inconsistency of the proposed addition of these elements from Bachand was pointed out during prosecution. The rejection was maintained on the contention that the operability of the proposed combination <u>is not completely foreclosed</u>.

Appellant respectfully submits that this is a legally incorrect view of the obviousness standard and represents an improper attempt to burden Appellant with showing that the proposed combination is <u>impossible</u>. Instead, Appellant submits that the obviousness standard is based on the objective perceptions and motivations of one of ordinary skill and the burden of proof lies with the Office. However, the rejection has never set forth why it would be obvious for one of ordinary skill to start comparing addresses that, to a large extent, Mann does not store. Accordingly, the mere fact that it is not impossible to practice the claimed subject matter in Mann's system does not imply that one of ordinary skill would find some affirmative suggestion or motivation to practice the claimed subject matter, as required for the Office to establish *prima facie* case of obviousness.

Further, even if Bachand and Mann were combined, one of ordinary skill in the art would still not arrive at the claimed subject matter. In this regard, Bachand is directed toward maintaining cache coherency in a multi-agent architecture. A cache is coherent when each agent uses the most current copy of data available to the system. *See Bachand*, para. [0003].

Each of Bachand's agents exchanges cache coherency messages (i.e., snoop responses) that identify whether other agents possess copies of requested data in the MESI scheme. *See Bachand*, para. [0009]. For the MESI scheme to operate, these cache coherency messages must be exchanged before the data is provided to a requesting agent, lest copies of data that are not current be used. *See Bachand*, para. [0009].

Bachand explicitly states that this is applicable to his technology. In particular, in Bachand's system, an external transaction queue compares the address of newly requested data with addresses of pending posted transaction data (i.e., transactions with addresses that have not yet been globally observed). *See Bachand*, para. [0025].

Appellant submits that Bachand's newly requested data address is not a fetched instruction address, as recited in claim 1. To begin with, the MESI scheme is designed to operate with data, not instructions. Further, Bachand and the MESI scheme is explicit in that any existing possession of data must be identified before another agent can establish possession. To contend otherwise (i.e., that Bachand's comparisons can be made after data is provided to agents), would render the entire MESI scheme inoperable to prevent cache incoherency. Bachand's comparison must therefore occur before data is provided to a requesting agent.

When this deficiency in Bachand was pointed out in the response filed Sept. 30, 2005, the rejection is understood to contend that Bachand's new data requests constitute a fetch command. *See January 17, 2006 Office action*, page 3. However, claim 1 recites the comparison of fetch<u>ed</u> instruction addresses, not new data requests.

Moreover, perhaps since Bachand deals with data requests rather than instructions, it is not surprising that Bachand fails to make any comparison of source addresses, such as performed at the second comparator recited in claim 1. In this, Bachand has the exact same deficiency as Mann, who does not store source addresses.

Moreover, the rejection of claim 1 is based on the contention that Bachand's single observation detection logic 246 somehow constitutes both a first comparator to compare a new branch target address corresponding to a loop in the first holding register to a stored branch target address in the first end register and a second comparator to compare a new branch source address corresponding to the loop in the second holding register to a stored branch source address in a first adjacent register. The alleged basis of this equivalence is the performance of successive comparisons by observation detection logic 246.

Appellant submits that the performance of successive comparisons by a single comparator does not transform the single comparator into multiple comparators. As recited in claim 1, the first comparator is to compare a new branch target address corresponding to a loop in the first holding register to a stored branch target address in the first end register, whereas the second comparator to compare a new branch source address corresponding to the loop in the second holding register to a stored branch source address in a first adjacent register. The comparators thus compare different data to achieve different comparisons results and are not the same single comparator.

Accordingly, even if Bachand were combined with Mann, one of ordinary skill in the art would still not arrive at the claimed subject matter.

Tanihara does nothing to remedy this deficiency in the combination of Mann and Bachand. In this regard, Tanihara describes a data shifting circuit of a central processor and has nothing to do with either trace buffer circuits or trace operations.

Accordingly, claim 1, and the claims dependent therefrom, are not obvious over Mann, Tanihara, and Bachand, individually or in combination. Appellant thus requests that the rejections of claims 1 and the claims dependent therefrom be withdrawn.

II: REJECTIONS OF CLAIMS 18, 27, THEIR DEPENDENCIES UNDER 35 U.S.C. § 103(a)

The rejections of claims 18 and 27 contend that one of ordinary skill could use Bachand's snoop blocking technique in Mann's debug interface to arrive at the claimed subject matter.

Appellant respectfully disagrees. As discussed above, Mann's trace compression relies upon screening of the information that is reported in trace data to limit the amount of stored data. In particular, Mann explicitly excludes addresses that are otherwise obtainable from a program image and certain disruptions in the program flow from reporting. Moreover, when disruptions in the program flow are reported, only a single target address or segment base address is recorded per disruption. Indeed, as best understood by applicant, Mann's trace buffer 200 never includes an address pair that corresponds to a loop, as recited in claims 18 and 27.

Against this backdrop, the rejection of claims 18 and 27 contend that, on the basis of Bachand's disclosure, it would be obvious for one of ordinary skill to add a compression operation that includes comparing the stored address pair to a new address pair in fetched instructions and setting a least significant bit of an address in the stored address pair in response to the new address pair matching the stored address pair, as recited in claims 18 and 27, to Mann's trace buffer 200.

Appellant respectfully disagrees. Mann's trace compression relies upon screening of the information that is recorded in trace buffer 200. As a result of this screening, few of Mann's trace entries even contain address values, much less the stored address pair as recited in claims 18 and 27. It defies common sense to conclude that one of ordinary skill would somehow be motivated to compare information that is not, in large part, stored.

When the inconsistency of the proposed addition of these elements from Bachand was pointed out, the rejection was maintained on the contention that the operability of the proposed combination is not completely foreclosed. However, as discussed above, the mere fact that it is not impossible to practice the claimed subject matter in Mann's system does not imply that one of ordinary skill would find some affirmative suggestion or motivation to practice the claimed subject matter, as required for the Office to establish *prima facie* case of obviousness.

Further, even if Bachand and Mann were combined, one of ordinary skill in the art would still not arrive at the claimed subject matter. In this regard, Bachand is directed toward maintaining cache coherency in a multi-agent architecture. In particular, cache coherency messages are exchanged before data is provided to a requesting agent and the dåta addresses identified in those messages are compared.

Appellant submits that Bachand's newly requested data addresses are not fetch<u>ed</u> instruction addresses, as recited in claims 18 and 27. The MESI scheme is designed to operate with data, not instructions. Further, Bachand's comparison must therefore occur before data is provided to a requesting agent.

Moreover, Bachand fails to make any comparison of stored address pairs, as recited in claims 18 and 27. In this, Bachand has the exact same deficiency as Mann, who does not store more than a single address when he stores anything at all.

Accordingly, even if Bachand were combined with Mann, one of ordinary skill in the art would still not arrive at the claimed subject matter. Thus, claims 18, 27, and the claims dependent therefrom, are not obvious over Mann and Bachand, individually or in combination. Appellant thus requests that the rejections of claims 18, 27, and the claims dependent therefrom be withdrawn.

The brief fee of $500 is enclosed. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: June 12, 2006

Scott C. Harris
Reg. No. 32,030

BY
JOHN F. CONROY
REG. NO. 45,485

Fish & Richardson P.C.
12390 El Camino Real
San Diego, California 92130
Telephone: (858) 678-5070
Facsimile: (858) 678-5099

SCH/JFC/jhg
10632620.doc

Appendix of Claims

1. A trace buffer circuit comprising:

a plurality of interconnected registers, including a first end register to input and output addresses of fetched instructions during a trace operation, a second end register, and a plurality of middle registers connected between said first end register and said second end register;

a write path to shift an instruction address in one of said plurality of interconnected registers by two registers toward the second end register on a write operation;

a first holding register;

a second holding register;

a first comparator to compare a new branch target address corresponding to a loop in the first holding register to a stored branch target address in the first end register;

a second comparator to compare a new branch source address corresponding to the loop in the second holding register to a stored branch source address in a first adjacent register, said first adjacent register being connected to the first end register on a read path; and

a compression indication circuit to generate a compression indicator in response to the new branch target address matching the stored branch target address and the new branch source address matching the stored branch source address.

2. The circuit of claim 1, further comprising:

a read path to shift the instruction address by one register toward the first end register on a read operation.

3. The circuit of claim 1, wherein the trace buffer operates as a first-in first-out (FIFO) register on the write operation and a last-in first-out (LIFO) register on the read operation.

4. The circuit of claim 1, wherein the instruction address comprises a 32-bit word.

5. The circuit of claim 4, wherein each of the first and second end registers and the plurality of interconnected registers comprise a 32-bit register.

6. The circuit of claim 5, wherein the plurality of interconnected registers comprise thirty-two registers.

7. The circuit of claim 5, further comprising:

a 64-bit write bus to write a 64-bit address pair to the first end register and an adjacent register on the write operation; and

a 32-bit read bus to read a 32-bit instruction address from the first end register on the read operation.

8. (Canceled)

9. The circuit of claim 1, wherein the compression indication circuit operates to set a least significant bit of the stored branch target address in response to the new branch target address matching the stored branch target address and the new branch source address matching the stored branch source address.

10. The circuit of claim 1, further comprising:

a second adjacent register in said plurality of registers, said second adjacent register being connected to the first adjacent register on the read path;

a third adjacent register in said plurality of registers, said third adjacent register being connected to the second adjacent register on the read path;

a third comparator to compare the new branch target address in the first holding register to a second stored branch target address in the second adjacent register; and

a fourth comparator to compare the new branch source address in the second holding register to a second stored branch source address in the third adjacent register,

wherein the compression indication circuit operates to generate a compression indicator in response to the new branch target address matching the second stored branch target address in the second adjacent register and the new branch source address matching the second stored branch source address in the third adjacent register.

11. The circuit of claim 10, wherein the compression indication circuit operates to set a least significant bit of the stored branch source address in the third adjacent register in response to the new branch target address matching the stored branch target address in the second adjacent register and the new branch source address matching the stored branch source address in the third adjacent register.

12. The circuit of claim 1, further comprising a valid bit buffer comprising:

a first end flip-flop to input and output valid bits from the valid bit buffer;

a second end flip-flop;

a plurality of interconnected flip-flops connected between said first end flip-flop and said second end flip-flop;

a write path to shift a valid bit in one of said plurality of interconnected flip-flops by two flip-flops to a downstream flip-flop on a write operation; and

a read path to shift the valid bit by one flip-flop toward an upstream flip-flop on a read operation.

13. A pipelined processor comprising:

a trace buffer circuit connected to the pipelined digital signal processor, said trace buffer circuit comprising:

a plurality of interconnected registers, including a first end register to input and output addresses of fetched instructions during a trace operation, a second end register, and a plurality of middle registers connected between said first end register and said second end register;

a write path to shift an instruction address in one of said plurality of interconnected registers by two registers toward the second end register on a write operation; and a read path to shift the instruction address by one register toward the first end register on a read operation.

14. The processor of claim 13, wherein the trace buffer operates as a first-in first-out (FIFO) register on the write operation and a last-in first-out (LIFO) register on the read operation.

15. The processor of claim 13, wherein the instruction address comprises a 32-bit word.

16. The processor of claim 15, wherein each of the first and second end registers and the plurality of interconnected registers comprise a 32-bit register.

17. The processor of claim 16, further comprising:

a 64-bit write bus to write a 64-bit address pair to the first end register and an adjacent register on the write operation; and

a 32-bit read bus to read a 32-bit instruction address from the end first register on the read operation.

18. A method comprising:

performing a trace operation including storing an address pair corresponding to a loop in fetched instructions in a trace buffer; and

performing a compression operation including

comparing the stored address pair to a new address pair in fetched instructions, and

setting a least significant bit of an address in the stored address pair in response to the new address pair matching the stored address pair.

19. The method of claim 18, further comprising:

discarding the new address pair in response to the new address pair matching the stored address pair.

20. The method of claim 18, further comprising:

storing the stored address pair in a first pair of registers; and

comparing the new address pair to the stored address pair.

21. The method of claim 20, further comprising:

writing the new address pair to the first pair of registers in response to the new address pair not matching the stored pair.

22. The method of claim 20, further comprising:

comparing the new address pair to a second stored address pair in a second pair of registers adjacent the first pair of registers;

setting a least significant bit of an address in the second stored address pair in response to the new address pair matching the second stored pair; and

writing the new address pair to the first pair of registers in response to the new address pair not matching the second stored pair.

23. The method of claim 22, further comprising:

discarding the new address pair in response to the new address pair matching the second stored address pair.

Claims 24-26. (Canceled)

27. An apparatus, including instructions residing on a machine-readable medium, for use in a trace buffer, the instructions operable to cause a machine to:

perform a trace operation including storing an address pair corresponding to a loop in fetched instructions in the trace buffer; and

perform a compression operation including

comparing the stored address pair to a new address pair in fetched instructions, and

setting a least significant bit of an address in the stored address pair in response to the new address pair matching the stored address pair.

28. (Original) The apparatus of claim 27, further comprising instructions causing the machine to:

discard the new address pair in response to the new address pair matching the stored address pair.

29. The apparatus of claim 27, further comprising instructions causing the machine to:

store the stored address pair in a first pair of registers; and

compare a new address pair to a stored address pair in the first pair of registers.

30. The apparatus of claim 29, further comprising instructions causing the machine to:

write the new address pair to the first pair of registers in response to the new address pair not matching the stored pair.

31. The apparatus of claim 29, further comprising instructions causing the machine to:

compare the new address pair to a second stored address pair in a second pair of registers adjacent the first pair of registers;

set a least significant bit of an address in the second stored address pair in response to the new address pair matching the second stored pair; and

write the new address pair to the first pair of registers in response to the new address pair not matching the second stored pair.

32. The apparatus of claim 31, further comprising instructions causing the machine to:

discard the new address pair in response to the new address pair matching the second stored address pair.

33. The method of claim 18, wherein said address in the address pair comprises a branch target address.

34. The method of claim 22, wherein said address in the second address pair comprises a branch source address.

Evidence Appendix

None

Related Proceedings Appendix

None

F&R 12-9-1999 **Checklist for APPEAL BRIEF** June 12, 2006

Client/Matter Number: 10559-292001

Client/Matter Name: P9299/ADI - A FIFO WRITE/LIFO READ TRACE BUFFER WITH SOFTWARE AND HARDWARE LOOP COMPRESSION

Sec	Att	PS	Check Items	
			PRE-FILING DETERMINATIONS	
☑	□	□	**Timing**	Notice of Appeal and, if applicable, the Final Office Action has been checked to confirm the due date was docketed correctly and is satisfied by this filing. This appeal brief is being filed: ● within 2 months after the Notice of Appeal ○ within 7 months after the Notice of Appeal with an extension of ___ months ○ within 6 months after the Final Office Action with an extension of ___ months
			REQUIRED FILING ENCLOSURES	
☑	□	□	**Caption**	(1) serial number, & (2) filing date have been checked for accuracy against information in the file.
☑	□	□	**Postcard**	(1) billing attorney's initials, & (2) list of all papers being sent and the pages of each.
☑	□	□	**Appeal Brief format**	(1) filed with **1 original**, (2) identifies filing date of Notice of Appeal or mailing date of Final Office Action, (3) lists all items being submitted, & (4) includes standard charges/credits statement
☑	□	□	**Appeal Brief items**	Appeal brief contains the following items under appropriate headings and in the order indicated: ☑ Real Party In Interest ☑ Related Appeals And Interferences ☑ Status Of Claims ☑ Status Of Amendments ☑ Summary Of Claimed Subject Matter ☑ Grounds of Rejection ☑ Argument ☑ Appendix Of Claims ☑ Appendix of Evidence (Appendix must be included even if there is no evidence. If no evidence, include the word "None" in the Appendix). ☑ Appendix of Related Proceedings (Appendix must be included even if there are no related proceedings. If no related proceedings, include the word "None" in the Appendix).
☑	□	□	**§1.17(f) Filing Fee**	● CHECK ○ DEPOSIT ACCOUNT CHARGE, indicated on Appeal Brief
☒ NA	□	□	**Ext. of Time**	● NONE REQUIRED ○ INCLUDED, with fee
☑	□	□	**Mail Certificate**	(1) includes 1st Class mail certificate, & (2) is signed and dated
☑	□	□	**Envelope**	Preprinted envelope or label is used, addressed to: MAIL STOP APPEAL BRIEF - PATENTS Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450
			TASKS AFTER ATTORNEY/AGENT CHECK	
□		□	**File copies**	(1) are complete & (2) include all signatures and dates
□			**Manual Docket**	Billing secretary's manual docket entry is updated
□			**Database Update**	Copy of postcard sent to Practice Services
□			**Folder Update**	File copy, tab, and updated table of contents are filed in prosecution folder

Checked By:

Secretary	Attorney/Agent	Practice Systems	Date
			6·12·06

PTO CHECK REQUEST

Date Requested	6/12/2006
Requested By	John F. Conroy/jhg
Matter Name	P9299/ADI - A FIFO WRITE/LIFO READ TRACE BUFFER WITH SOFTWARE AND HARDWARE LOOP COMPRESSION
Matter ID	10559-292001
Payable To	Commissioner of Patents and Trademarks
Check No.	
Check Date	
Check Amount	$500

Category		Amount	Client Share	Partner Share	Office Share
999 Other (explain below) Appeal Brief		$500.00	$500.00	$	$
		$	$	$	$
		$	$	$	$
1 Month	□ Small Entity-$60 □ Large Entity-$120	$	$	$	$
2 Month	□ Small Entity-$225 □ Large Entity-$450	$	$	$	$
3 Month	□ Small Entity-$510 □ Large Entity-$1020	$	$	$	$
4 Month	□ Small Entity-$795 □ Large Entity-$1590	$	$	$	$
5 Month	□ Small Entity-$1080 □ Large Entity-$2160	$	$	$	$

Billing Attorney	Scott C. Harris
Charge Office[1]	
Explanation	

Responsible Attorney Signature		Date	6.12.06
Approval Signature[2]	AQMinkus SDPreston	Date	

[1] If charged to office, the remainder of the form must be completed. If an extension fee, continuation, divisional, or fwc filing fee, or notice of appeal fee is charged to the office, the responsible partner shares the cost and must sign this request. In cases of combined attorney/client delay, the partner is expected to bill a reasonable amount to the client. Any remaining amount will be divided between the partner and the office according to the applicable percentage (0% of month 1 fee + 25% of month 2 fee + 50% of month 3 fee +50% of month 4 fee).

[2] All client-related office charges greater than $200 must be approved by AQMinkus and SDPreston. If advance approval is not possible, the person requesting the check must submit a write off request to SDPreston as soon as possible. Forms are found in Public Folders.